intel®

ARTICLE
REPRINT

AR-274

March 1983

The chip that refreshes itself

An 8k x 8 memory chip with onchip refresh and control circuitry offers designers the best aspects of both dynamic and static RAMS.

John J. Fallin
Joseph P. Altnether
William H. Righter
Intel Corporation

ORDER NUMBER: 230652-001

THE CHIP THAT REFRESHES ITSELF

An 8k x 8 memory chip with onchip refresh and control circuitry offers designers the best aspects of both dynamic and static RAMs.

by John J. Fallin, Joseph P. Altnether, and William H. Righter

The ideal microprocessor memory has three major characteristics: ease of use, flexibility, and low cost. Unfortunately, all of these characteristics are seldom available in a single device. Thus, memory system design involves compromises. Matching memory component characteristics to desired design parameters rarely results in a perfect fit.

To optimize memory design, the designer must set priorities in choosing between static random access memory (SRAM) and dynamic random access memory (DRAM). Neither choice completely supplies all the beneficial characteristics. SRAMs are easy to use, but high cost, limited density, and high power consumption limit their application to under 64k bytes. DRAMs are suitable for large memory arrays (above 64k bytes) where density and low power are important. In addition, the refresh overhead of large memory systems is spread over a large amount of memory. Between the two extremes represented by static and dynamic RAMs, however, is a performance zone where all three ideal characteristics (low cost, flexibility, and ease of use) are required.

Low cost can be achieved with a DRAM cell. A major factor determining the cost of a memory device is the physical area occupied by the memory cell. Small cell size allows small die size, which results in more die per wafer. The net result is lower cost. DRAMs achieve lower unit costs because they use a single transistor for the memory cell. Each SRAM cell, on the other hand, requires six transistors. In addition, the DRAM cell provides the benefits of low power and high density. The remaining ideal characteristics (flexibility and ease of use) demand more innovative approaches than are presently available.

A microprocessor memory system using DRAMs consists of three major elements: microprocessor, memory, and controller. To gain ease of use, the controller must be incorporated into either the microprocessor or the memory. By including the memory control in the microprocessor, a simple memory device can be built in which the cost of the control logic is distributed over the entire memory system. Sadly, this approach has inherent faults. Because refresh is derived from microprocessor timing, any operation that suspends or stretches the timing must be carefully analyzed to ensure that it does not violate system refresh timing. Examples of such time stretching operations are extended wait states, hold operations in direct memory access, or the single-step operations important in system debugging. Moreover, placing the burden of refresh on the microprocessor can reduce processor performance by suspending operation to service memory refresh.

Another technique for refresh control is to incorporate most of the refresh mechanism on the RAM chip, leaving the most difficult portion—the arbiter—to the system designer. This type of RAM is known as a

John J. Fallin is currently an applications engineer at Intel Memory Products, 2111 NE 25th, Hillsboro, OR 97123. He is responsible for iRAM applications. Mr Fallin has a BSEE from Oregon State University.

Joseph P. Altnether is the technical marketing manager at Intel Memory Products and is involved in the technical marketing of Intel iRAMs. Mr Altnether has a BSEE from St Louis University.

William H. Righter is also an applications engineer at Intel Memory Products. He is responsible for the graphics applications of dynamic RAMs. Mr Righter is currently working on a BSEE from the University of Portland.

pseudo- or quasistatic RAM. Statistical techniques or special features of a particular central processing unit (CPU) may be used to accomplish refresh, though it is not always guaranteed. The system must be thoroughly analyzed to ensure proper refresh timing.

The third memory refresh method incorporates all of the memory control logic, including arbitration logic, into the RAM chip. Thus, all DRAM cell control is contained on the same piece of silicon, creating a complete integrated memory system on a chip. This concept is called an integrated RAM (iRAM.) Combining the best features of dynamic and static RAM, the iRAM satisfies the ideal microprocessor memory requirements—a DRAM storage cell for low cost, a Joint Electron Device Engineering Council 28-pin package configuration conforming to universal flexibility standards, and ease of use because of onchip control. (See the Panel.)

Intel's 8k x 8 iRAM contains all the elements needed for complete memory control. Recognizing the need for both synchronous and asynchronous refresh, Intel offers the 2187 iRAM with synchronous refresh for special applications and the 2186 with asynchronous refresh for general purpose designs. With onchip refresh, the 2186 needs no external stimulus or control. As a result, memory is autonomous; like an SRAM it can be left alone with power applied and still retain data. Any operation that suspends or stretches the system timing has no effect on refresh operation. However, because refresh is asynchronous with the microprocessor, a memory request can occur during a refresh cycle. In this case, the iRAM signals the system microprocessor that a delay will occur in the cycle.

Device operation

The 2186/2187 performs four types of cycles: read, write, false memory, and refresh. It is important to note that chip enable ($\overline{\text{CE}}$) is an edge-triggered, not a level-triggered, input. On the 2186/2187, an access cycle is requested for every high to low transition of $\overline{\text{CE}}$. Care must be taken to ensure that a new access cycle is not requested before the previous cycle is completed. This would violate memory cycle time (TELEL) and would jeopardize data integrity. A minimum precharge ($\overline{\text{CE}}$ high time—TEHEL) must also be guaranteed. The violation of $\overline{\text{CE}}$ high time would most likely occur in systems where noise spikes can occur on chip enable.

Functioning like a clocked static RAM, iRAM addresses are latched off the external address bus on the falling edge of $\overline{\text{CE}}$. This feature is useful in many designs because it saves the designer one or two transistor-transistor logic packages. In contrast to the trailing edge write of the SRAM, the iRAM requires a leading edge write. Further, the iRAM permits three different types of access cycles. Depending on the active time of $\overline{\text{CE}}$ and its relation to output enable ($\overline{\text{OE}}$) or write enable ($\overline{\text{WE}}$), a long mode cycle, a pulsed mode cycle, or a false memory cycle (FMC) can be performed.

A long mode cycle is similar to a fully static RAM cycle in that $\overline{\text{CE}}$ remains valid throughout the entire cycle. In the pulsed mode cycle, the iRAM operates as a clocked static RAM and $\overline{\text{CE}}$ is active only at the beginning of the cycle. When $\overline{\text{CE}}$ becomes active, without an $\overline{\text{OE}}$ or $\overline{\text{WE}}$, the iRAM performs an FMC and terminates on the rising edge of $\overline{\text{CE}}$. This is useful during byte write operations of 16-bit microprocessors. In this case, a 16-bit word is selected (two iRAMS), and only one receives a write pulse to perform byte write. The other selected iRAM performs an FMC.

Assuming a refresh is not in progress when the cycle begins, an access cycle is initiated on the high to low transition of $\overline{\text{CE}}$. After activating $\overline{\text{CE}}$, addresses are latched from the external bus and data are presented to the bus. Data will remain at the bus as long as $\overline{\text{OE}}$ remains active, independent of the state of $\overline{\text{CE}}$.

Any operation that suspends or stretches the system timing has no effect on refresh operation.

If a refresh cycle is in progress at the time $\overline{\text{CE}}$ goes low, a deferred cycle occurs. In this case, ready (RDY) will respond by going low within a given time (TELRL) of $\overline{\text{CE}}$ going low. After the refresh cycle and part of the access cycle complete, RDY will return to a high state. At a specified time after this (TRHQV), valid data will become available at the data input/output (I/O) outputs.

It should be noted that $\overline{\text{OE}}$ can remain active for an unlimited period of time, and data will remain on the bus even though internal refresh cycles continue to be performed. This allows operation in systems using single-stepping hardware debug, since wait states can be inserted at will. RDY does not respond under these conditions.

For the sake of clarity, assume that a refresh cycle is not in progress at the time of $\overline{\text{CE}}$'s falling edge. A write cycle begins in the same way as a read cycle, with addresses latched on the falling edge of $\overline{\text{CE}}$. For a pulsed mode, $\overline{\text{WE}}$ must go low within a specified time of the falling edge of $\overline{\text{CE}}$ (TELWL). If this specification time is not met, an FMC occurs and thus, no write. For a long mode, TWLEH must be met to ensure a write given setup time before (TDVWL) and a given hold time (TWLDX) after the leading edge of $\overline{\text{WE}}$.

If a refresh cycle is already in progress at the onset of a write cycle, the RDY will respond at a given time (TELRL) after $\overline{\text{CE}}$'s falling edge. Data will still be latched into the device on the falling edge of $\overline{\text{WE}}$, but the actual write to the array will not occur until after the refresh cycle is completed. RDY responds during write cycles to prevent cycle timer (TELEL) violations. As in a read cycle, $\overline{\text{CE}}$ and $\overline{\text{WE}}$ can remain active for an indefinite period to accommodate single-step type operation. An FMC occurs when $\overline{\text{CE}}$ becomes active but neither $\overline{\text{OE}}$ nor $\overline{\text{WE}}$ becomes active. An FMC is a valid mode of operation and also acts like a row address strobe (RAS) only refresh, in which the row selected by the seven external row addresses is refreshed.

Internal vs external refresh

The 2186 internal refresh is completely automatic, requiring no external stimulus; an internal timer provides refresh requests. If an access cycle is requested during a refresh cycle, the 2186 will respond by outputting a RDY low. For applications requiring maximum performance, the 2187 allows external generation of refresh signals. A high to low transition on the refresh enable ($\overline{\text{REFEN}}$) input of the 2187 will initiate a refresh cycle. After starting a refresh cycle, one cycle time (TELEL) must be allowed before attempting another access or refresh cycle. Deferred access cycles are not allowed on the 2187, as it has no arbitration circuitry. If $\overline{\text{REFEN}}$ is held low for at least one timer period, the internal timer will begin to time out, and the 2187 will maintain refresh with no outside intervention.

Inside the iRAM

Included in the 5-V iRAM device are a dynamic array, an arbiter, a refresh address counter, and a refresh timer along with complete control and precharge circuitry. The 2187 differs from the 2186 only in the refresh timer and arbiter control circuitry. The diagram illustrates the interrelation of these iRAM elements.

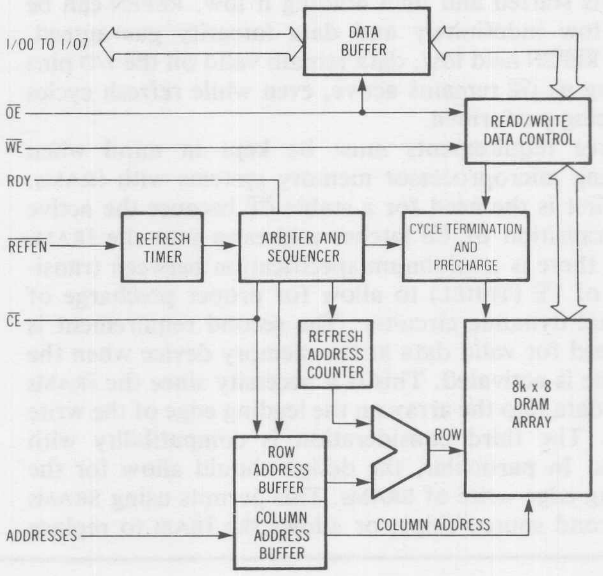
Arbiter
The arbiter that determines the priority sequence of two or more asynchronous inputs is the most significant element of the 2186 internal refresh circuitry. In the 2186, the two inputs to the arbiter are an external access cycle request and an internal refresh cycle request. When either of these requests is made, the arbiter decides whether the cycle will proceed immediately or be delayed. For example, if an access cycle is in progress at the time of an internal refresh request, the refresh cycle will be delayed until after the access cycle is completed. Conversely, if an access cycle is requested while the 2186 is performing a refresh cycle, the access cycle will be delayed. Here the 2186 will respond with a RDY low output, instructing the device that more time must be allowed. In the limit, both cycle requests can occur simultaneously. Therefore, arbitration becomes necessary along with the simple state gating as outlined.

Array
The memory array, designed with direct random access memory (DRAM) storage cells, is fabricated using an N-channel double-layer polysilicon gate process. The array features polysilicon word lines and folded metal bit lines that provide high common mode rejection.

Refresh timer
Refresh peripheral circuitry is included on the device to preserve the integrity of the data in the DRAM array. A refresh timer provides refresh cycle requests at appropriate intervals. The timer is designed to track with both process variations and temperature so as to guarantee proper refresh over all specified ranges.

Row address counter
When the internal refresh cycle is granted, refresh addresses are provided by a refresh address counter. This counter contains the 7-bit address of the next row to be refreshed and is incremented after each refresh cycle. Control logic within the peripheral circuitry handles the multiplexing of external memory addresses and refresh addresses during appropriate cycles.

Device pinout
The 2186/2187 comes in Joint Electron Device Engineering Council compatible 28-pin socket. Pin 1 (labeled "$\overline{\text{CNTRL}}$") becomes the RDY output on the 2186, or the $\overline{\text{REFEN}}$ input on the 2187. Pin functions are described in the Table.



iRAM Pin and Signal Definitions

Pin	Description/Function
A_0 to A_{12}	Address inputs. These inputs provide the addresses needed to select one of 8192 bytes. Addresses appearing on these inputs are latched into the device on the high to low transition of $\overline{\text{CE}}$.
I/O_0 to I/O_7	Data input/output. These bidirectional pins receive data to be written to the device and output data during a read cycle.
$\overline{\text{CE}}$	Chip enable input. A high to low transition on this pin latches addresses and initiates an access cycle.
$\overline{\text{OE}}$	Output enable input . During a read cycle this input turns on the data output buffers.
$\overline{\text{WE}}$	Write enable input. Data are latched into the device on the leading (falling) edge of this signal.
↓RDY (2186 only)	This output becomes active to signify a delay in the cycle. It is open drain, allowing the wire ORing of several 2186 RDY outputs.
↓$\overline{\text{REFEN}}$ (2187 only)	This input allows for external refresh control. A high to low transition of $\overline{\text{REFEN}}$ causes a refresh cycle to be initiated. Holding $\overline{\text{REFEN}}$ low causes asynchronous timer operation.
V_{CC}	This input supplies operating voltage to the device. V_{CC} is specified at 5 V ± 10%.
GND	Ground input for the device.

After $\overline{\text{REFEN}}$ returns high from this state, a minimum amount of time (TRFHEL) must be allowed before the next falling edge of $\overline{\text{CE}}$ or $\overline{\text{REFEN}}$ in order to complete any refresh cycles initiated by the timer. This mode of operation is called power-down refresh. The 2187 also supports a single-step mode of operation, which is accomplished by strobing $\overline{\text{REFEN}}$ low after an access cycle is started and then holding it low. $\overline{\text{REFEN}}$ can be kept low indefinitely and data integrity guaranteed. With $\overline{\text{REFEN}}$ held low, data remain valid on the I/O pins as long as $\overline{\text{OE}}$ remains active, even while refresh cycles are being performed.

Three requirements must be kept in mind when building microprocessor memory systems with iRAMs. The first is the need for a stable $\overline{\text{CE}}$ because the active low transition of $\overline{\text{CE}}$ latches addresses into the iRAM. Also, there is a minimum specification between transitions of $\overline{\text{CE}}$ (TEHEL) to allow for proper precharge of internal dynamic circuitry. The second requirement is the need for valid data at the memory device when the $\overline{\text{WE}}$ line is activated. This is a necessity since the iRAMs write data into the array on the leading edge of the write pulse. The third consideration is compatibility with SRAMs. In particular, the design should allow for the trailing edge write of SRAMs. This permits using SRAMs as second source chips, or allows the iRAM to replace read only memory (ROM) or erasable programmable read only memory (EPROM) during system debug stages.

Following are applications that exemplify the techniques for interfacing the asynchronous 2186 and the synchronous 2187 iRAMs to various microprocessors. Although the designs can be simpler, additional circuitry is included to provide memory site compatibility with SRAMs and EPROMs.

5-MHz 8088 processor application

The first example involves a 5-MHz 8088 microprocessor configured to a bank of 2186 iRAMs. It runs in the maximum mode without wait states for normal memory access cycles, except when RDY is activated due to internal refresh. The schematic diagram is shown in Fig 1. The iRAM chip enable circuit is a simple cross-coupled latch that provides an active low enable signal ($\overline{\text{E}}$) synchronized with address latch enable (ALE). This enable signal, along with latched status bit S2, is used to enable the 74S138 address decoder to provide stable chip enable signals ($\overline{\text{CE}}$0 to $\overline{\text{CE}}$7) to the 2186 iRAMs. The memory write control ($\overline{\text{MWTC}}$) output from the 8288 bus controller provides for both leading and trailing edge write conditions.

The basic operation of the $\overline{\text{CE}}$ circuit is as follows: early in the CPU cycle, ALE goes high, then low, clearing the cross-coupled latch and driving $\overline{\text{E}}$ high. $\overline{\text{E}}$ remains

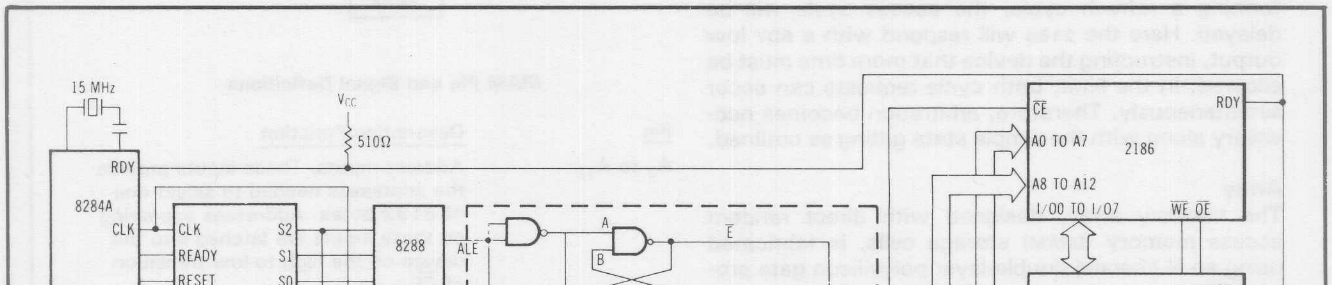


Fig 1 Schematic for iRAM use with 8088 microprocessing unit running in maximum mode. To ensure proper startup, all iRAM control inputs must be inactive for 100 μs after V_{CC} reaches spec.

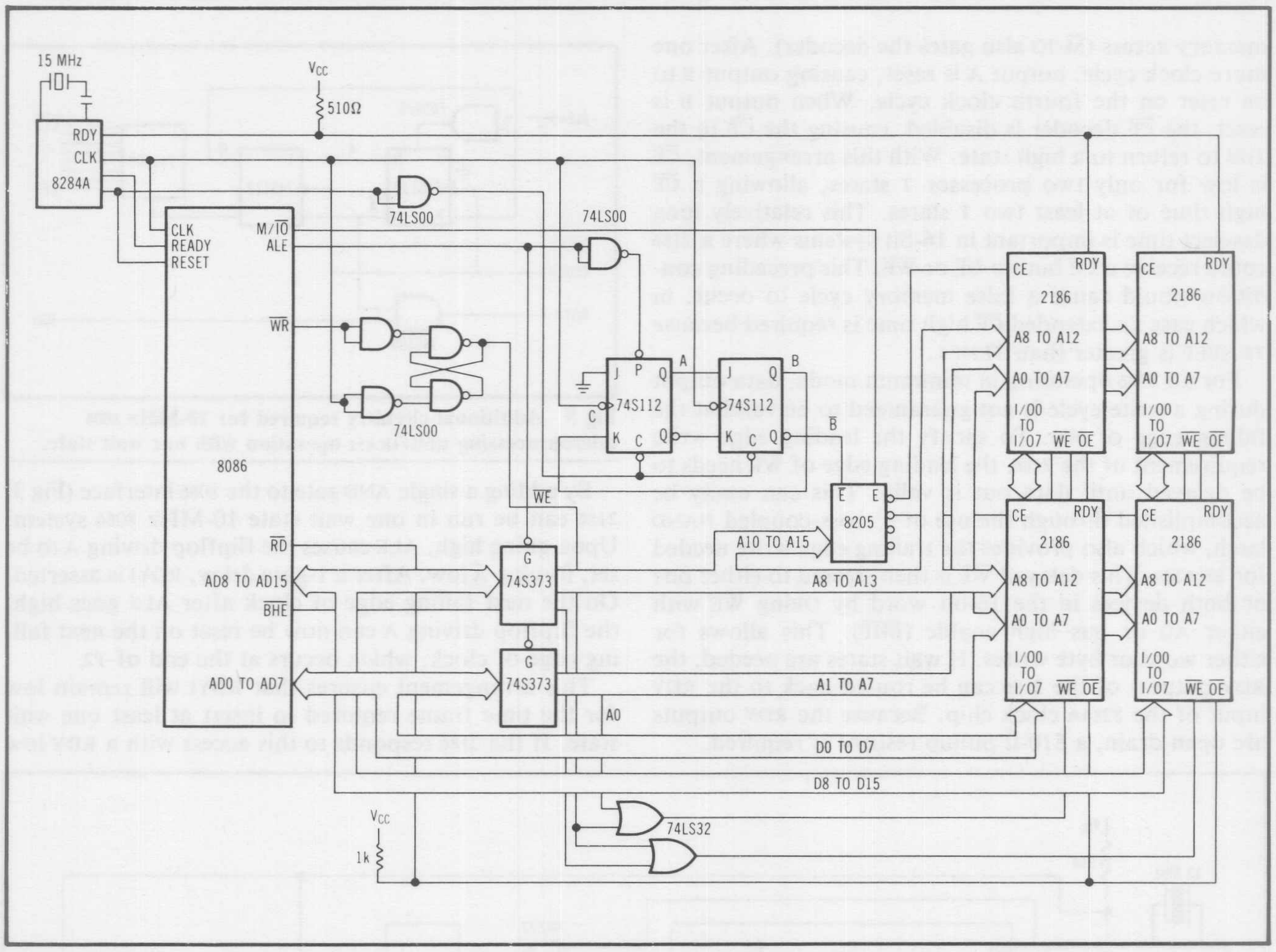


Fig 2 Schematic for iRAM use with 8086 microprocessing unit operation in minimum mode. NAND latch in processor's $\overline{\text{WR}}$ line ensures adequate delay of data to iRAM as well as supplying trailing edge WR signal required by conventional SRAMs.

high at least as long as ALE holds the clear condition of the latch. Due to the skew of the falling edge of ALE with the system clock, two possible decoder enable timing sequences exist. The first occurs when ALE returns to a low state before the falling edge of the system clock. In this case, the latch remains cleared and $\overline{\text{E}}$ remains high. On the falling edge of the system clock, the latch is set, driving $\overline{\text{E}}$ low and enabling the decoder (depending on the state of S2). The other timing sequence occurs when ALE goes low after the falling edge of the system clock. In this case, when ALE goes low, the $\overline{\text{E}}$ signal is immediately driven low as a result of the system clock (low) input on the "set" side of the latch. The memory cycle completes and early into the next cycle when ALE goes high, $\overline{\text{E}}$ returns to a high state. This clearing action occurs independently of the state of the clock.

The net result is the enabling of the 74S138 decoder after its address inputs have stabilized. Thus, a transient-free chip enable is supplied to the iRAMs. The balance of the memory system is wired in a straightforward manner. The $\overline{\text{OE}}$ signal for the memory array is connected directly to the read control ($\overline{\text{RD}}$) signal of the 8088 processor. Bidirectional data lines of the memories are connected to the processor's AD0 to AD7 lines. CPU addresses are latched by the 74S373s two gate delays after the falling edge of ALE. Although the iRAMs do not require the address latches, they are included for completeness. A typical system needs to latch the addresses from the multiplexed bus for interfacing to SRAMs, EPROMs, and so on. The address lines feed the memory array via the 74S373 flow-through latches. All of the RDY lines of the memory array connect to a RDY input of the 8284A clock generator in an OR configuration. A 510-Ω pullup resistor is used for stability.

Note that S2 is latched into a 74LS74 flipflop on the rising edge of ALE. This is important because, during certain CPU operations such as the execution of the halt instruction, the status bits are not guaranteed to remain valid until the falling edge of ALE. To guarantee proper power-up of the 2186, all control inputs must remain inactive for 100 μs after V_{CC} reaches specification. This is accomplished by tying RESET to the clear input of the M/IO flipflop. A pullup resistor on the $\overline{\text{OE}}$ is also required because the RD line on the 8088 goes into a high impedence state during RESET.

iRAMs in an 8086 based system

The interface requirements for an 8086/2186 system (Fig 2) are similar to those for an 8088. The $\overline{\text{CE}}$ generation circuitry consists of two JK flipflops, arranged as a 4-state sequencer. This sequencer makes its first transition on the rising edge of ALE, when sequencer output A is set. On the next falling edge of the system clock, sequencer output B is set, enabling the 8205 decoder if the cycle is a memory access ($\overline{M}$/IO also gates the decoder). After one more clock cycle, output A is reset, causing output B to be reset on the fourth clock cycle. When output B is reset, the $\overline{CE}$ decoder is disabled, causing the $\overline{CE}$ to the 2186 to return to a high state. With this arrangement, $\overline{CE}$ is low for only two processor T states, allowing a $\overline{CE}$ high time of at least two T states. This relatively long deselect time is important in 16-bit systems where a 2186 could receive a $\overline{CE}$ but no $\overline{OE}$ or $\overline{WE}$. This preceding condition would cause a false memory cycle to occur, in which case an extended $\overline{CE}$ high time is required because TEHELF is greater than TEHEL.

For an 8086 operating in minimum mode, data output during a write cycle is not guaranteed to be valid at the falling edge of WR. To satisfy the leading edge write requirement of the 2186, the leading edge of $\overline{WR}$ needs to be delayed until data out is valid. This can easily be accomplished through the use of a cross-coupled NAND latch, which also provides the trailing edge write needed for SRAMs. This delayed $\overline{WE}$ is then steered to either one or both devices in the 16-bit word by ORing $\overline{WE}$ with either AO or bus high enable ($\overline{BHE}$). This allows for either word or byte writes. If wait states are needed, the RDY outputs of the 2186 can be routed back to the RDY input of the 8284A clock chip. Because the RDY outputs are open drain, a 510-Ω pullup resistor is required.

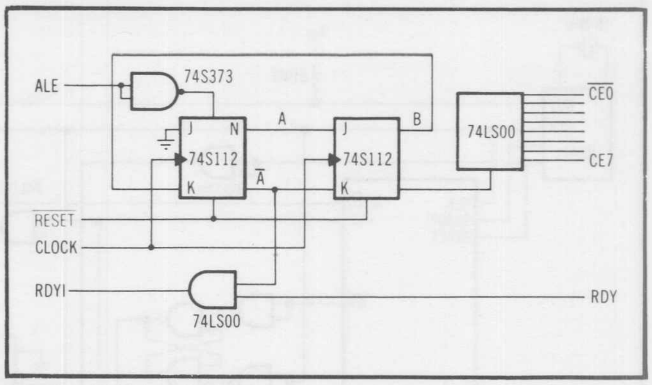


Fig 3 Additional circuitry required for 10-MHz 8086 microprocessing unit/iRAM operation with one wait state.

By adding a single AND gate to the 8086 interface (Fig 3) 2186 can be run in one wait state 10-MHz 8086 system. Upon going high, ALE causes the flipflop driving A to be set, forcing $\overline{A}$ low. After a 1-gate delay, RDYI is asserted. On the next falling edge of clock after ALE goes high, the flipflop driving A can now be reset on the next falling edge of clock, which occurs at the end of T2.

This arrangement ensures that RDYI will remain low for the time frame required to insert at least one wait state. If the 2186 responds to this access with a RDY low

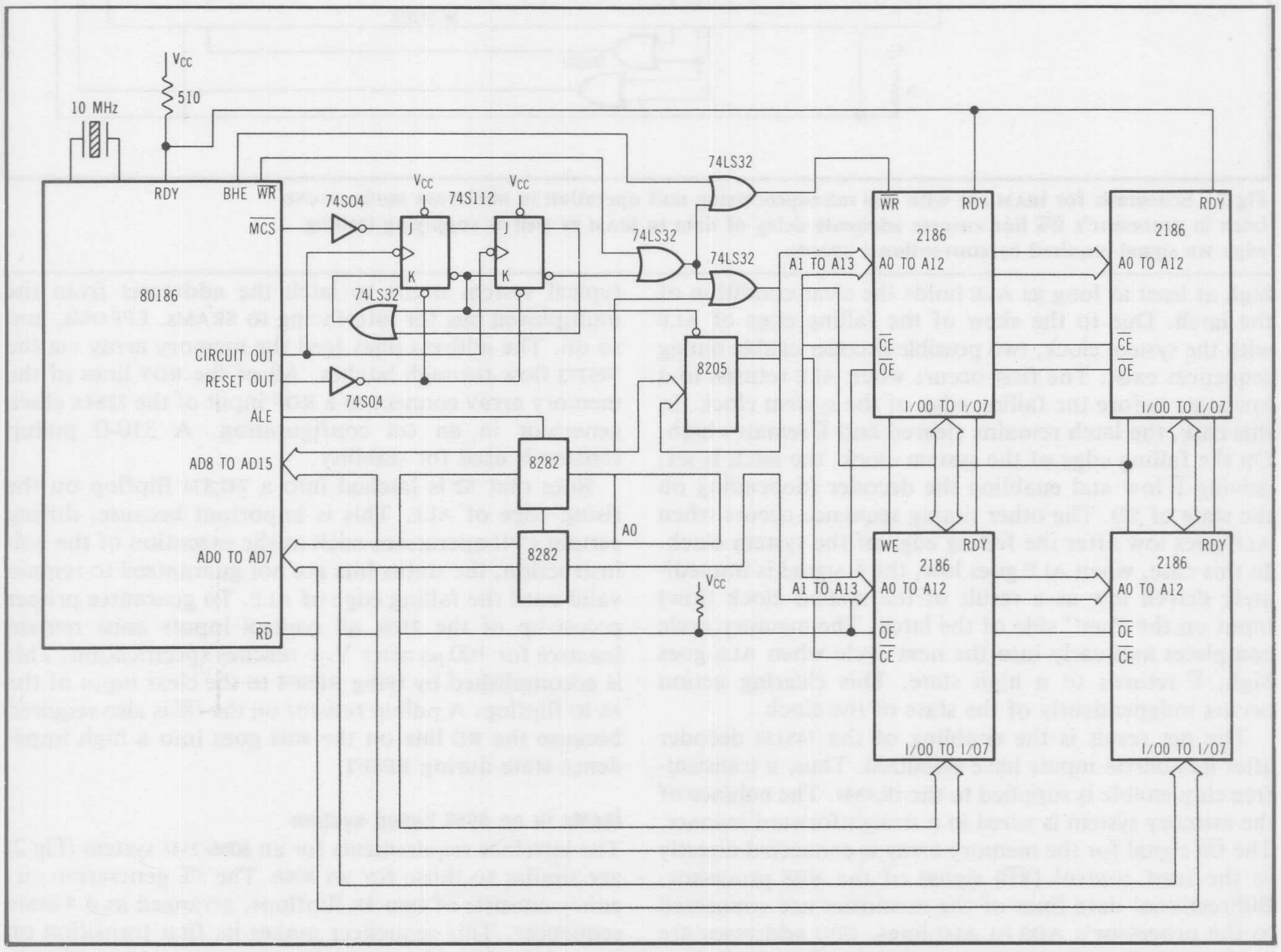


Fig 4 Interface circuitry required for iRAM operation with 5-MHz 80186 microprocessing unit. This system, based on popular iAPX 86 family of components, provides users with programmable memory chip selects for blocked memory applications.

condition, the trailing edge of RDYI will be pushed out to insert more wait states as needed. The number of wait states that can be inserted will never exceed six—assuming a maximum RDY low time of 575 ns.

Fig 4 shows the circuitry required to interface the 2186 with Intel's 80186 high integration 16-bit processor. The 80186 is an integration of 10 common iAPX 86 components, including an enhanced 8086-2 CPU, a clock generator, a local bus controller, and an interrupt controller. The clock speed of this configuration is 5 MHz. The 80186's bus structure is much like that of the 8086, but there are some useful additions, including programmable memory chip select (MCS) outputs. These outputs can be programmed to become active for a user-selected block of memory. In Fig 4, one of four available mid-range MCSs would be programmed to become active in the memory space allocated to the iRAM. This MCS signal is then used to initiate a count sequence with the two JK flipflops. The two flipflops' outputs provide both the $\overline{CE}$ and the properly positioned $\overline{WE}$ (for leading edge writes) to the iRAMs.

FMCs, in which the iRAM receives a $\overline{CE}$ but no $\overline{OE}$ or $\overline{WE}$, will occur any time a single byte write occurs. If this is the case, the $\overline{CE}$ high time requirement becomes somewhat longer than that required for a read or write cycle. The $\overline{CE}$ circuitry used was selected with this consideration in mind because of the long $\overline{CE}$ high time it guarantees.

Synchronous iRAMs in an 8088 system

One way to create a synchronous design using the 8088, operating in maximum mode with the 2187 iRAM, is to use a status decoder to generate a refresh signal during an opcode fetch cycle. The following example assumes that the 2187 iRAM is used for data store only, so that the iRAM can be sent a refresh strobe during the time the 8088 is executing an instruction fetch from some other portion of memory (ie, EPROM, ROM). It also assumes that opcode fetches occur often enough to meet the 128 refreshes/2-ms refresh specification.

In the circuit shown in Fig 5, it is evident that the 2187 and the 8088 are interconnected with a minimal amount

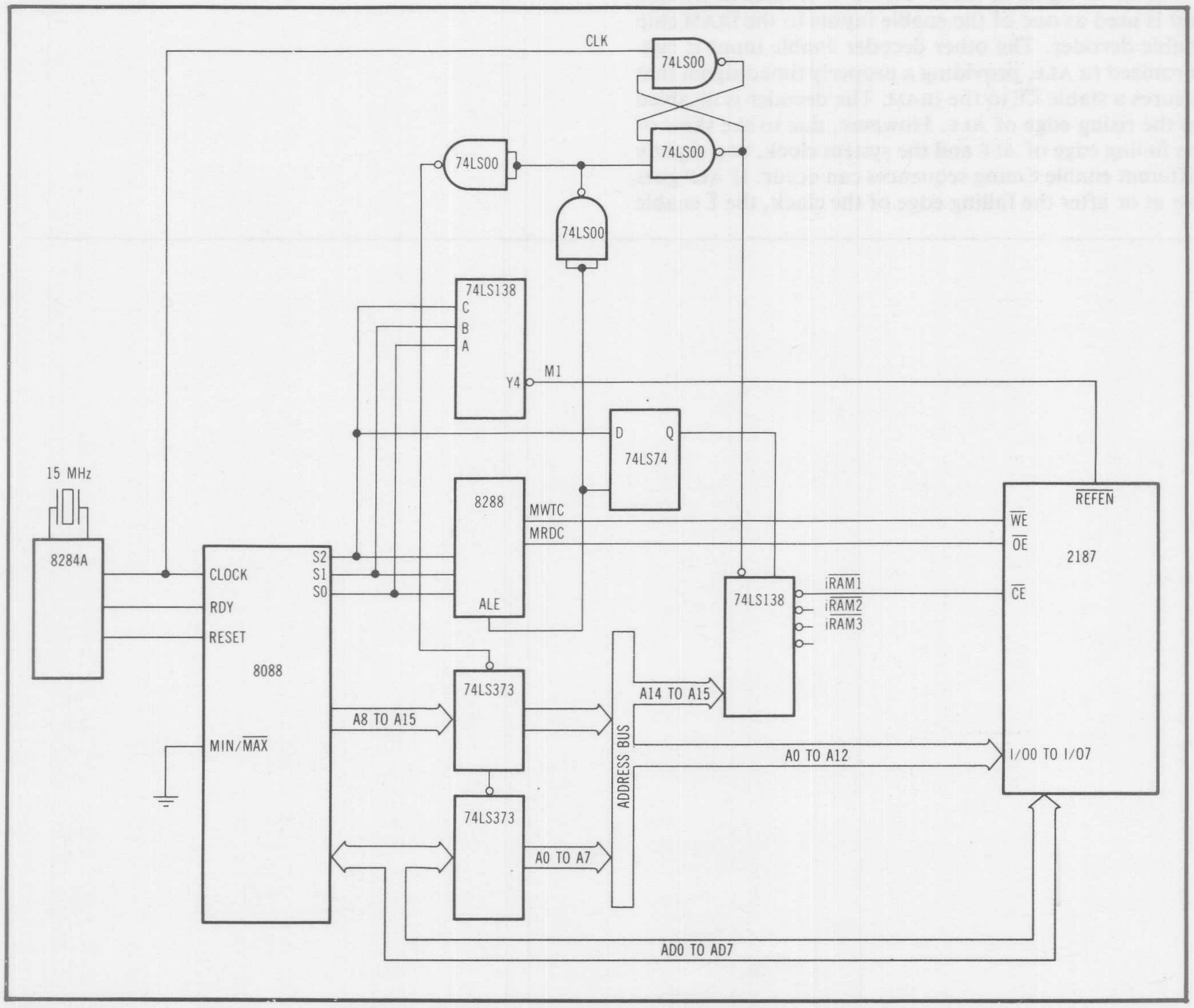


Fig 5 System diagram of 2187/8088 synchronous system. iRAM refresh occurs during 8088 instruction fetch cycle and is generated by external status decoder. Minimal transistor-transistor logic interface circuitry is required.

Combining memory and control together on a single piece of silicon... satisfies all the requirements for microprocessor memory.

of transistor-transistor logic interface circuitry. The decoded 8088 signal M1 is connected to the $\overline{\text{REFEN}}$ input of the iRAM. This connection generates a refresh strobe to the 2187 every time the 8088 performs an opcode fetch. The 8288 bus controller generates memory read commands and memory write commands that are properly timed for all memory requirements. Thus, these signals may connect directly to the $\overline{\text{WE}}$ and $\overline{\text{OE}}$ control lines of the iRAM without special conditioning circuitry. The 8-bit multiplexed address-data bus of the 8088 is connected directly to the I/O 0 to 7 lines of the iRAM. The low order addresses A0 to A7 are latched by ALE at the 74LS373 latch and , together with lines A8 to A12, form the iRAM device address.

The M/IO signal is status bit S2. It is latched by ALE and is used as one of the enable inputs to the iRAM chip enable decoder. The other decoder enable input is synchronized to ALE, providing a properly timed signal that ensures a stable $\overline{\text{CE}}$ to the iRAM. The decoder is disabled on the rising edge of ALE. However, due to the skew of the falling edge of ALE and the system clock, two slightly different enable timing sequences can occur. If ALE goes low at or after the falling edge of the clock, the $\overline{\text{E}}$ enable line is immediately activated and enables the decoder. Note that when RESET is low, REFEN will be forced low. This guarantees proper 2187 power-up. This circuit runs at 5 MHz without wait states.

Combining memory and control together on a single piece of silicon, the iRAM satisfies all the requirements for microprocessor memory. Its ease of use, flexibility, and low cost make it an attractive memory alternative. By using it in some of the applications demonstrated, engineers will find that the iRAM can help them meet the majority of both present and future memory system design needs.